The Free UCS Outline Fonts Project — an Attempt to Create a Global Font

Primož Peterlin
University of Ljubljana
Faculty of Medicine, Institute of Biophysics
Lipičeva 2, SI-1000 Ljubljana, Slovenia
primoz.peterlin@biofiz.mf.uni-lj.si
http://biofiz.mf.uni-lj.si/~peterlin/

Abstract

In February 2002, the Free UCS (Universal Character Set) Outline Fonts project (http://savannah.gnu.org/projects/freefont/) was started. Exercising the open-source approach, its aim is to provide a set of free Times-, Helvetica- and Courier-lookalikes available in the OpenType format, and progressively cover the complete ISO 10646/Unicode range. In this stage of the project, we focus mainly on two areas: collecting existing fonts that are both typographically and license-wise (i.e., GNU GPL) compatible and can be included to cover certain parts of the character set, and patching up smaller areas that are not yet covered. Planned future activities involve typographic refinement, extending kerning information beyond the basic Latin area, including TrueType hinting instructions, and facilitating the usage of fonts with various applications, including the TEX/Ω typesetting system.

Résumé

Le projet de *fontes vectorielles libres pour UCS* (http://savannah.gnu.org/projects/freefont/) a démarré en février 2002. À travers l'*Open Source*, le but de ce projet est de fournir un ensemble de clones libres de Times, Helvetica et Courier, disponibles dans le format OpenType, et couvrant progressivement l'ensemble des caractères d'ISO 10646/Unicode. À ce stage du projet nous nous concentrons sur deux domaines : la collection de fontes existantes qui soient compatibles avec notre projet, aussi bien du point de vue typographique que du point de vue de la licence (GNU GPL), qui puissent être intégrées pour couvrir certaines parties de l'ensemble de caractères ; et d'autre part, faire des ajouts de petites régions qui n'ont pas encore été couvertes. Nos activités futures prévoient le perfectionnement typographique, le crénage au-delà de l'alphabet latin, l'incorporation de *hints* TrueType, et le support de plusieurs applications, dont aussi les systèmes TEX et Ω.

Introduction

The aim of the Free UCS Outline Font project is to provide a standardised set of glyphs which make a harmonised design despite including glyphs from different scripts (Latin, Cyrillic, Greek, Armenian, etc.). It is clear that this requires compromises at the cost of typographic finesse (see [11] for a discussion of typographic compromises regarding the Greek alphabet), yet the end result must look acceptable for general use, including electronic mail, world-wide web, and text editors.

While this is clearly not the first attempt to create a typeface covering glyphs beyond Latin, previous attempts are not as numerous as one might imagine. Bigelow [5] quotes *Romulus* by Jan van Krimpen from 1931 as one of the first examples. Another example is Николай Николаевич Кудряшов [Nikolaĭ Nikolaevich Kudryashov]'s *Encyclopaedia* (Кудряшовская енциклопедическая [Kudryashovskaya entsiklopedicheskaya]) family (1960–1974), originally designed for the third edition of the Great Soviet Encyclopaedia which contains thousands of glyphs, including Cyrillic, Latin, and Greek letters in serif and sans-serif styles, as well as other special signs and symbols. Most popular typefaces such as Stanley Morison's Times Roman, Max Miedinger's Helvetica, Adrian Frutiger's Univers and Eric Gill's Gill Sans were originally designed for the Latin alphabet and later, as they gained popularity, extended to Greek and Cyrillic alphabets.

A typeface covering the whole ISO 10646/Unicode range has to be designed flexibly enough to allow addition of scripts not belonging to the Western typographic tradition. The task is not easy, and it is not surprising that so far there are very few aesthetically satisfying typographic solutions. Among those one has to mention Bigelow and Holmes's *Lucida Sans Unicode* [5] and Haralambous and Plaice's Ω [12]. Other solutions like *Arial Unicode* [16] and James Kass's compendium *Code2000* [14] are too varied in style to form a unified typeface.

Design issues

Historical and cultural context. It is clear that a typeface aiming to cover "all the scripts of the world" can not rely on historical styles (e.g., Renaissance, Baroque, etc.) known from Western typography, as most of the world has not experienced these periods in the evolution of typography. Trying to extend them to non-European scripts[1] is as inappropriate as, say, trying to design a Latin alphabet in Kufi style.

Even the apparently logical division into seriffed and sans-serif typefaces is eurocentric. To see why, one only has to think of the origin of Latin capital letters. Serifs are an invention of Roman stone-carvers — a smaller stroke perpendicular to the main stroke was added to provide a uniform smooth finishing of the stroke. Here, the visual effect followed the available technology. The modulated stroke — another ancient Roman typographic invention — is an opposite example, where the technology followed the visual effect [13]. The need to strengthen vertical strokes arose once the Romans started to erect monuments of monumental proportions, where the inscriptions were no longer in the eye-height. While the rain took away the paint from the vertical strokes, dirt and dust accumulated in the horizontal strokes, which thus appeared optically heavier. *Physical* broadening of vertical strokes was introduced as a compensation, in order for the horizontal and vertical strokes to look *optically* equivalent. The Roman technological innovations survived for 2 000 years in European typography. However, in countries where the letters were painted with brush onto silk or carved with a needle onto a palm leaf rather than carved into the stone with mallet and chisel, such typographic development never happened.

Similar concerns about eurocentrism are valid for differing the upright and the italic forms, or for that matter, even between *majuscules* and *minuscules*, capital and small letters. Even differing between upright and slanted forms is questionable, as slanted forms make no sense in, say, most native Asian scripts.

The least questionable seems to be the differences based on the weight of the typefaces, which appears to be almost universal. The only exception known to the author is Ethiopic,[2] where the words were traditionally emphasised by printing them in another colour (red in religious texts, blue in imperial decrees) or by underlining them or enclosing them in ovals.

Legibility. A typeface which includes non-Latin scripts offers an opportunity for exploring the "universal" parameters of legibility. While there has been a wealth of publication on legibility centred on Latin script [20, 22, 9, 15], exploring the factors like weight, serif vs. sans-serif faces, x-height, capitalisation etc., it is clear that such differences are minor compared to the differences in shape between Latin and, say, Hebrew, Arabic, Devanagari or Tamil, which nevertheless provide roughly the same level of legibility. Lacking comparative cross-script studies, the experiments in legibility remain in the realm of the typographer's intuition. On the brighter side, making multi-script typefaces available, albeit not perfect, is a step towards the world where such cross-cultural studies will be easier to achieve.

Methodology

The basic idea behind the free UCS outline fonts project was to collect various available free outline fonts, covering single national scripts, and to compile them into a large font using the ISO 10646/Unicode coded character set [21], taking into consideration typographic and legal compatibility, and filling in the missing areas on the way. The whole development was planned to be carried in the open-source manner, with many developers using a central repository.

The general requirement for technical realisation was that typefaces need to be available as scalable vector fonts. The actual technical realisation (PostScript Type 1 [1] uses cubic Bézier splines, while TrueType [3] uses quadratic ones) was considered secondary, because at least in principle it is possible to transform the fonts from one form to another. In reality, though, no known transformations is completely lossless — kerning and hints are usually the most volatile.

Licenses of used sources. We anticipated that our result will contain glyphs from many different sources. Thus, special attention was given to the license under which a font is released. The license we looked for should allow redistribution, modification and distribution of modified font files. Many free and open-source licenses fulfil this requirement. As the URW++ core PostScript fonts, the Ω-Serif and some other major sources were released under the GNU General Public License [10], we adopted it for our project as well. The license itself is suited for programs rather than typefaces and its application to fonts may be legally dubious, even though, say, PostScript Type 1 fonts are perfectly legal programs written in PostScript. We were not able to find any license in the same spirit pertaining specifically to fonts, though. It may be worth seeing the final license of the Bitstream Vera fonts, which were announced in January 2003 to be soon available under a license in the open-source spirit.

We thus limited our search to fonts which were not only typographically compatible, but also specifically released under the GNU GPL. In some cases, where

1. Even though historical styles differ throughout Europe, Europe is nevertheless treated here as a historical and cultural unity when contrasted with the rest of the world.

2. Daniel Yacob, personal communication.

the fonts were released under less clearly defined terms ("free", "public domain" etc.), we contacted the authors and asked for permission to use their work under the terms of the GNU GPL. Most authors agreed, and while none disagreed, some of the emails remained unanswered. In such cases we were unable to confirm whether the recipient received our email at all. These fonts still wait to be included in the free UCS outline font collection.

Choice of typefaces. Even though Bigelow and Holmes claim that their primary reason for choosing a sans-serif font is because it carries least historical and cultural associations [5, p. 1003], and despite the expressed concerns about eurocentrism, we decided to develop in parallel three different families, modelled after Times Roman, Helvetica and Courier, and specifically derived from URW++ typefaces Nimbus Roman No. 9, Nimbus Sans and Nimbus Mono. Reflecting the nature of the project, we dubbed them as Free Serif, Free Sans and Free Mono.

Aside from covering three different letterforms — one monospaced and two proportional, one with modulated strokes and another with unmodulated strokes — the primary reason for choosing these three typefaces was their ubiquity. Since they have been extremely popular for many decades, and present in the desktop publishing world-wide for over twenty years, they inspired numerous local designs around the world, where non-Latin glyphs were designed specifically to blend with one of the above-named typefaces. While we realise that many of these designs introduce typographic practice alien to a traditional local design, we also recognise that the designs done by native speakers reflect the local typographic knowledge and its evolving typographic rules.

Tools. While we paid special attention to the licensing of the used sources, no particular attention was given to the licensing issues of the tools used, as the font tools generally don't imply the licenses under which the fonts are released. Still, we ended up using exclusively open-sourced tools. The choice was clearly influenced by the fact that most participants of the project use Linux, and while none of the numerous popular font editors from other platforms have been ported to this platform, we have some excellent free tools available.

PfaEdit Without George Williams's excellent font editor PfaEdit,[3] this project would not have been possible at all. PfaEdit is a visual font editor that allows copying and modifying glyphs and most other things expected from a font editor, reads and writes most popular outline font formats, and has proved to be stable enough to allow working on large glyph sets. Even though the program was mature enough to be used as a tool back in late 2001, the pace of its development has not diminished over the past year, and its author is very responsive to bug reports.

Choosing PfaEdit as our main tool also influenced the decision of the native font format used in the free UCS outline fonts project. Since we use a CVS repository for the bookkeeping of font additions and modifications, we were looking for a format suitable to differential changes. A format was considered suitable if local changes in typeface design remained localised in the font file rather than propagating themselves across the file. This requirement ruled out compiled binary formats, as well as PostScript Type 1 `eexec` encoding. PfaEdit's native format SFD, similar to Adobe's BDF format in its structure, proved to be a suitable choice. The fonts are thus archived in the SFD format, and PfaEdit is used to create fonts in other outline formats.

PfaEdit's only major drawback may be its weak support for TrueType instructions or "hints". Since TrueType is not the native format used by the free UCS outline font project, we are looking either for automated generation of TrueType hints, or for separate files with TrueType instructions which could be compiled together with the glyph outlines in the SFD format to produce a hinted TrueType font. Any solution requiring manual modification of binary TrueType fonts is not acceptable, as TrueType fonts are created automatically each time from the SFD sources. While PfaEdit will probably eventually produce acceptable TrueType hints in an automated way, other possibilities have to be investigated in the meantime.

TrueType tools Rogier van Dalen's TrueType tools[4] might be the proper solution for the TrueType hinting problem. Van Dalen's approach uses a separate file with TrueType instructions written in a high-level language not unlike C, which can be compiled together with an unhinted TrueType font file to produce a hinted TrueType font file. If this approach is adopted, each TrueType font file will be automatically produced from two source files: the SFD file containing the glyph outlines and the TTI file containing the TrueType instructions for hinting. At the moment of this writing, the main concern is the estimated amount of work needed for producing TrueType hinting instructions in a manual way.

TTX While the SFD format fits present needs just fine, the future might require distributing font sources in some other open format. For this purpose we

3. Since renamed to FontForge: `http://fontforge.sourceforge.net/`

4. `http://home.kabelfoon.nl/~slam/fonts/`

are considering TTX, the TrueType to XML converter.[5] Given the pace at which PfaEdit currently evolves, it may well be that PfaEdit itself will support XML before we will actually feel the need for it.

Towards ISO 10646/Unicode compliance. As of version 3.2, Unicode defines 95 156 encoded characters. Taking into account three families (Free Serif, Free Sans and Free Mono), two weights (normal and bold) and two shapes (regular and italic/oblique), this means designing almost 1.2 million glyphs for the free UCS font project. Being a volunteer project, free UCS outline fonts project grows mainly in a non-systematic way — when a suitable set of glyphs is found or donated, we add it to the font. Often, a wider community can benefit from scratching one's own itch, e.g. creating a set of APL glyphs [7, 8].

Still, we felt the need to add the glyphs in a more systematic way in parallel with the spontaneous growth of the project. For that purpose, we use the multilingual European subsets as defined by Comité Européen de Normalisation [6] as a guideline:

- MES-1, a Latin repertoire based on ISO/IEC 6937:1994 (335 characters)
- MES-2, a Latin, Cyrillic, and Greek repertoire based on ENV 1973:1996 (1062 characters)
- MES-3, a repertoire needed to write all the languages of Europe and transliterate between them. MES-3A is a script-based, non-fixed collection, while MES-3B is a fixed subset of 2819 characters.

Character-to-glyph translation. As the ISO 10646/Unicode standard encodes *characters* rather than *glyphs*, it is clear that for acceptable rendering of many languages more glyphs than those corresponding to Unicode characters are needed. This is particularly important for rendering Indic languages and Syriac which contain numerous ligatures not encoded in the ISO 10646/Unicode standard.

During the 1990's, several initiatives for creating "smart fonts" were started, i.e., fonts being not only containers of glyphs but also incorporating some logical rules for glyph substitution, glyph position, contextual rendering, etc. Among those were Apple Advanced Typography [2], SIL Graphite [18] and Adobe/Microsoft OpenType [17]. From these three, OpenType seems to be grabbing the largest market share.

Results

The project was conceived in December 2001 and was approved for hosting on the Savannah Web site[6] in February 2002. As of March 2003, a total of 17 794 characters are encoded. We are just a couple of dozen glyphs short of MES-1 compliance in Free Serif and Free Sans, while Free Mono is already MES-1 compliant, and approximately 3 500 glyphs short of MES-2 compliance.

Kerning. Aside from the character set compliance and the already mentioned TrueType hinting problem, there remain some other tasks. One of them is kerning, which is currently present only in the Latin portions of the font. Kerning non-Latin scripts requires typographic knowledge which we currently do not possess. We believe that the open nature of the project will eventually attract somebody with the necessary expertise who is willing to contribute.

Discussion

The open-source development model has not been previously tested for font development. While it may be arguable whether such a development model can be applied at all to creation of works of art, it is also worth noting that most contributors have a technical and scientific background, and probably none of them perceives him- or herself as an artist.

While there is no doubt that a skilled typographer would be able to design a multi-script font vastly exceeding what we have done in typographic beauty and consistency, we must not overlook the fact that designing 1 200 000 glyphs is probably beyond the capability of a single person, even if we neglect the financial part of such an endeavour.

Free UCS outline fonts project and TEX. So far, the Free UCS outline fonts project has taken from the TEX community more than it has repaid. Thanks to Péter Szabó and his TEXtrace program (`http://www.inf.bme.hu/~pts/textrace/`) [19], it is easy to transform Metafonts into PostScript Type 1 fonts which can be edited by PfaEdit. Karel Piška demonstrated the technique for Indian Metafonts during the TUG 2002 conference in Thiruvananthapuram, India.

Even though the free UCS outline fonts were designed primarily for screen use, we can certainly raise the question of using the fonts for typesetting with TEX or Ω. As of March 2003, this would require extensive work, such as splitting the fonts into smaller fonts containing no more than 256 glyphs. However, the current development [4] promises that there might be a more direct way of using OpenType fonts with Ω in the future.

Acknowledgments

I would like to thank the following authors who contributed to the project by allowing their work to be included in the free UCS outline fonts project. In alphabetical order: Berhanu Beyene, Daniel Shurovich Chirkov,

5. `http://www.letterror.com/code/ttx/`
6. `http://savannah.gnu.org/projects/freefont`

Prasad A. Chodavarapu, Vyacheslav Dikonov, DMS Electronics, Valek Filippov, Shaheed R. Haque, Yannis Haralambous, Angelo Haritsis, Jeroen Hellingman, Maxim Iorsh, Mohamed Ishan, Manfred Kudlek, Harsh Kumar, Sushant Kumar Dash, Olaf Kummer, Noah Levitt, Jochen Metzinger, Anshuman Pandey, Hardip Singh Pannu, Thomas Ridgeway, Young U. Ryu, Virach Sornlertlamvanich, M.S. Sridhar, Sam Stepanyan, URW++ Design & Development GmbH, Frans Velthuis, and the Wadalab Kanji Committee.

References

[1] Adobe Systems Incorporated. *Adobe Type 1 Font Format*. Addison-Wesley, 1990. Also available on the Web: `http://partners.adobe.com/asn/developer/PDFS/TN/T1_SPEC.PDF`.

[2] Apple Computer, Inc. Apple advanced typography. Available on the Web: `http://developer.apple.com/fonts/`, 1994.

[3] Apple Computer, Inc. TrueType reference manual. Available on the Web: `http://developer.apple.com/fonts/TTRefMan/`, 1999.

[4] Gábor Bella, Anish Mehta, and Yannis Haralambous. Adapting Omega to OpenType fonts. In *EuroTEX 2003* (this volume), 2003.

[5] Charles Bigelow and Kris Holmes. The design of a Unicode font. *Electronic Publishing*, 6(3):999–1015, 1993.

[6] CEN. Information technology – multilingual European subsets in ISO/IEC 10646-1. Technical Report CWA 13873:2000, Comité Européen de Normalisation, Technical Committee 304, 2000. Also available on the Web: `http://www.evertype.com/standards/iso10646/pdf/cwa13873.pdf`.

[7] Phil Chastney. An APL Unicode font. *Vector*, 16(1):75–85, 1999.

[8] Phil Chastney. APL Unicode font – extended. *Vector*, 17(3):139–142, 2001.

[9] Rudi W. de Lange, Hendry L. Esterhuizen, and Derek Beatty. Performance differences between Times and Helvetica in a reading task. *Electronic Publishing*, 6(3):241–248, 1993.

[10] Free Software Foundation. GNU General Public License, version 2. Available on the Web: `http://www.gnu.org/copyleft/gpl.html`, 1991.

[11] Yannis Haralambous. From Unicode to typography, a case study: the Greek script. In *Proceedings of 14th International Unicode Conference*, pages b.10.1–b.10.36, Boston, 1999.

[12] Yannis Haralambous and John Plaice. Ω, a TEX extension including Unicode and featuring Lex-like filtering processes. In *EuroTEX Proceedings*, pages 154–167, Gdańsk, 1994.

[13] Georges Jean. *L'écriture, mémoire des hommes*. Gallimard, Paris, 1987.

[14] James Kass. Code2000. Available on the Web: `http://home.att.net/~jameskass/`, 1998.

[15] Ole Lund. *Knowledge construction in typography: the case of legibility research and the legibility of sans serif typefaces*. PhD thesis, The University of Reading, Department of Typography & Graphic Communication, 1999.

[16] Microsoft Corporation. Arial Unicode MS. Available on the Web: `http://office.microsoft.com/downloads/2000/aruniupd.aspx`, 2000.

[17] Microsoft Corporation. OpenType specification. Available on the Web: `http://www.microsoft.com/typography/otspec/`, 2001.

[18] SIL International. Graphite. Available on the Web: `http://graphite.sil.org/`, 1997.

[19] Péter Szabó. Conversion of TEX fonts into Type 1 format. In *Proceedings of EuroTEX 2001 conference, Kerkade, September 24–27*, pages 192–206, 2001.

[20] Miles A. Tinker. *Legibility of Print*. Iowa State University Press, Ames, 1963.

[21] The Unicode Consortium. *The Unicode Standard Version 3.0*. Addison-Wesley, Reading, Massachusetts, 2000.

[22] Bror Zachrisson. *Studies in the legibility of printed text*. Almqvist & Wiksell, Stockholm, 1965.